

Document number: P0365R0

Date: 2016-05-30

Project: Programming Language C++, SG14, SG1, SG6, SG7, EWG, LEWG

Authors: Michael Wong

Reply to: michael@codeplay.com

Report on SG14, a year later and future directions

[Introduction](#)

[2015](#)

[2016](#)

[Status](#)

[Future Meetings](#)

[Future Directions](#)

[References](#)

Introduction

SG14 was officially started in 2015 Lenexa meeting after N4456 was presented to help serve the interest of the games, real-time, low-latency, graphics and embedded system. This paper serves to summarize recent changes, latest status of various papers, and papers of tracking interest to SG14, as well as future directions. It will not expand on specific proposal details. For that, please join the reflector and/or telecons.

This process was started with a CPPCON 2014 question from the games community during one of the panel I was on, about how to better support games for C++. As the question came from many Canadian games company at the conference, I organized an impromptu BoF to gather common interest. This gathering evolved further afterwards in late 2014/early 2015 led by Sean Middleditch with an online google forum on cxx-real-time discussion where common interests was discussed. At the same time, Guy Davidson also added github repository to help the group stage code and papers. These events and discussion culminated in N4456.

The common interest of the group was constrained resources, real-time graphics, low-latency requirements which were commonly intersected by games developers,

financial/trading/ simulations, and embedded devices. The uniqueness of this group was that it was one of the first SG that specifically represented named industry. Given that one of the issues from previous attempts by this group to connect with C++ was that they could not attend C++ Standard meetings due to their tight deadlines, the way it was to work was to have C++ Standard representatives who could regularly attend the meeting to act as a proxy for between meeting discussions/telecons of this community, such that their proposals can be represented by these volunteers.

This was augmented with actual F2F meetings where this group regularly do meet, which for gamers were CPPCON and GDC. Since then, we have also identified STAC and JAX financial software conferences as candidates for F2F meetings. In future, we would also consider embedded chip conferences to meet embedded programmers.

The group was chartered into existence by Herb Sutter on the Saturday lunch meeting before the final plenary to be called SG14 with the name focused on low-latency and chaired by me.

2015

In 2015, the SG was inaugurated with the following reflector and github and generally monthly telecon meetings hosted by Michael, now using a Codeplay Webex. Some of these telecons are grouped with SG 1 and during C++ Standard meetings, the two groups sit together as many proposals intersect both groups. However, proxies also visit all the other groups as they champion particular proposals. All the meetings are extremely well-attended with telecons averaging 10-20 and F2F meetings averaging 50, because this is a group with a name that industry can identify. The group often act as pre-screen for proposals, and can throw its support behind proposals that it likes.

- Reflector
 - <https://groups.google.com/a/isocpp.org/forum/?fromgroups=#forum/sg14> 
- Code and proposal Staging area
 - <https://github.com/WG21-SG14/SG14> 

While SG14 serves many small proposals that can be moved immediately, especially if they cross SG1,6,or7, SG14 also have mandate to consider major long-pole items such as heterogenous device support, vector, light-weight exception handling, affinity, and any issues relating to low-latency wherever they interest this community. These other major design proposals may appear as a TS.

The group met F2F at CPPCON 2015 in a full day SG14 meeting, resulting in a number of proposals reviewed and subsequently proxied at the Kona C++ Standard meeting. In addition to the usual rigorous paper reviews, this meeting featured an open discussion on what to do about exception handling brainstorm in which many games industry participants try to reconcile a way to create a light-weight exception handling mechanism to augment the current mechanism, in a way that would satisfy games programming interest. This has now resulted in a paper P0364r0.

In Kona, we presented 8 papers and the minutes can be seen here:

<https://groups.google.com/a/isocpp.org/forum/#!topic/sg14/-Dw-jYbhlAc>

We have 8 papers upstreamed from the meeting for Kona

P0037R0 Fixed point real numbers John McFarlane LEWG SG14/SG6: Lawrence Crowl

Continue with integration with Lawrence's proposal

P0038R0 Flat Containers Sean Middleditch LEWG SG14: Patrice Roy

Strong guidance to proceed

P0039R0 Extending raw_storage_iterator Brent Friedman LEWG SG14: Billy Baker

No strong interest

P0040R0 Extending memory management tools Brent Friedman LEWG SG14: Billy Baker

Strong support to forward into C++17

P0041R0 Unstable remove algorithms Brent Friedman LEWG SG14: Billy Baker

Mild consensus against proceeding

P0048R0 Games Dev/Low Latency/Financial Trading/Banking Meeting Minutes 2015/08/12-2015/09/23
Michael Wong SG14: none

P0059R0 Add rings to the Standard Library Guy Davidson LEWG SG14: Michael

Strong consensus to continue work

P0130R0 Comparing virtual functions Scott Wardle, Roberto Parolin EWG SG14: Michael

Did not discuss.

P0125R0 std::bitset inclusion test methods: Michael/Walter

Proceed to wording

SG14 also offered comments on the following proposals, some of which do not have papers as yet.

GPU Accelerator support

[P0069R0](#) A C++ Compiler for Heterogeneous Computing

We actually intend to study the design of C++AMP, OpenMP Accelerator/OpenACC, OpenCL, OpenGL and Vulkan for a GPU accelerator design to support gamers. We plan to review this paper under a co-located SG14 session which can run with SG1, but we are interested in taking this work further.

Allocators

[P0089R0](#) Quantifying Memory-Allocation Strategies

Coroutines and games

[P0054R0](#) Coroutines: reports from the fields

[P0055R0](#) On Interactions Between Coroutines and Networking Library

[P0070R0](#) Coroutines: Return Before Await

[P0071R0](#) Coroutines: Keyword alternatives

[P0073R0](#) On unifying the coroutines and resumable functions proposals

[P0099R0](#) A low-level API for stackful context switching

Intrusive containers

likely/unlikely

function multiversioning

low latency for financial/trading

matrix operations

SIMD vectors

[P0076R0](#) Vector and Wavefront Policies

Reflection

Numerics:

[P0106R0](#) C++ Binary Fixed-Point Arithmetic

There was also an evening session of SG14 was conducted which further refined the light-weight exception handling brainstorming while presenting it to a wider committee group.

We finished 2015 with a telecon specifically on co-routines and games based on Gor's co-routine proposals, where some of the ideas from expected and co-routine can be used in a light-weight EH proposal.

In 2015, I also attended Meeting C++ where there is a Embedded programming track, where SG14 started connecting with the interest of Embedded computing. SG14 is increasingly finding common cases with the embedded community, and in a recent talk in Europe, I was contact by several Embedded specialists and professor who wish to as SG14 to help. Specifically, the common themes that is shared between the 2 SGs are a wish to have compile time, static based features (like constexpr, reflection) and a wish to reduce the cost of exception Handling/ RTTI.

2016

In 2016, we kicked off with several telecons focused on unjamming vector/simd proposals as they are of deep interest for games programmers. We also started focusing on the other main long-pole item from Games and Embedded programmers, and that was discussing the various industry implementation models to support Massive Dispatch for Heterogeneous Device. The telecons had presentations from Nvidia's Agency, AMD's HCC, Khronos SYCL, and LSU's HPX.

We also discussed with Daniel Gutson who has been serving as the defacto-chair of the pseudo SG for Embedded devices. The group remains a pseudo group as it does not have a consistent presence at C++ Standard. They have been interested in and put forward proposals of the following:

P0132R0	Non-throwing container operations
N3986	Adding Standard support to avoid padding within structures
N3990	Adding Standard Circular Shift operators for computer integers
N4049	0-overhead-principle violations in exception handling

N4234	0-overhead-principle violations in exception handling - part 2
-----------------------	--

N4226	Apply the <code>[[noreturn]]</code> attribute to main as a hint to eliminate global object destructor calls
-----------------------	---

In addition, this group has been working on `constexpr` constructor memory allocation, non-contiguous vector, as well as EH Lite.

After some discussion, all of the Embedded devices was merged with SG14.

The other significant event of 2016 was the drive to integrate financial/trading/HFT industries. As such I have been attending STAC conferences and giving talks on C++17 status, SG14 status, and participating in a panel on what HFT's need from C++. We also held SG14 meetings at the local trading firms, hosted by Neil Horlock of Credit Suisse in London, Tom Rodgers and Neve Liber in Liber in Chicago. This group has specific interests in:

- a. Heterogeneous device support (Michael, Harttmu
- b. EH lite (see below)
- c. CPU/cache/memory affinity (Neil)
- d. memory allocation (Guy Davidson)

In the 2016 JAX meeting, SG14 proxied or followed the following papers:

P0249R0	Input Devices For 2D Graphics	Brett Searles
P0267R0	A Proposal to Add 2D Graphics Rendering and Display to C++,	Michael McLaughlin
P0037R1	Fixed point real numbers	John McFarlane
P0040R1	Extending memory management tools	Brent Friedman
P0059R1	Add rings to the Standard Library	Guy Davidson, Arthur O'Dwyer
P0203R0	Considerations for the design of expressive portable SIMD vectors	Mathias Gaunard

- P0230R0 SG14 Games Dev/Low Latency/Financial Meeting Minutes Michael
2015/10/14-2015/02/10 Wong
- P0232R0 A Concurrency ToolKit for Structured Paul McKenney, Michael
Deferral/Optimistic Speculation Wong, Maged Michael
- P0233R0 Hazard Pointers: Safe Reclamation Maged M. Michael,
for Optimistic Concurrency Michael Wong
- P0234R0 Towards Massive Parallelism(aka Michael Wong, Hartmut
Heterogeneous Kaiser, Thomas Heller
Devices/Accelerators/GPGPU)
support in C++
- P0235R0 A Packaging System for C++ Guy Somberg, Brian
Fitzgerald
- P0236R0 Khronos's OpenCL SYCL to support Michael Wong, Andrew
Heterogeneous Devices for C++ Richards, Maria Rovatsou,
Ruyman Reyes
- P0237R0 On the standardization of Vincent Reverdy, Robert
fundamental bit manipulation J. Brunner
utilities

P0193R0 Where is Vectorization in JF Bastien, Hans Boehm
C++?

Shortly afterwards, at GDC 2016, we had again a full day SG14 meeting hosted by JF Bastien of Google in which the morning was devoted to paper and JAX reviews:

- Sean Middleditch sean@middleditch.us Wargaming P0038R0 Flat Containers
- Arthur O'Dwyer arthur.j.odwyer@gmail.com y P0059R1 Add rings to the Standard Library
- Marco Foco marco.foco@gmail.com fixed point
- John Mcfarlane john@mcfarlane.name P0037R1 Fixed point real numbers
- Billy Baker Billy.Baker@flightsafety.com Flight Safety P0040R1 Extending memory management tools
- JF Bastien jfb@google.com Google P0193R0 Where is Vectorization in C++?
- Michael Wong fraggamuffin@gmail.com ISOCPP y P0233R0 Hazard Pointers: Safe Reclamation for Optimistic Concurrency
- Scott Wardle swardle@gmail.com EA P0130R0 Comparing virtual functions
- Lawrence Crowl lawrence.crowl@gmail.com P-106 C++ Binary Fixed Point
- Guy Somberg/Brian Fitzgerald bfitz@blizzard.com Activision/Blizzard P0235R0 A Packaging System for C++

The afternoon was devoted to general discussions on:

- 2.3.1 C++ Jacksonville meeting update
- 2.3.2: Sg6 updates
- 2.3.3: Sg1 updates
- 2.3.4: LEWG updates

- [2.3.5 Light weigh exceptions \(expected, status, disappointment\) \(Lawrence Crowl/ Paul Hampson\)](#)
 - [Lawrence Crowl lawrence.crowl@gmail.com](#) P0262R0 A Class for Status and Optional Value Handling Disappointment in C++ SG6 Updates
 - [Paul Hampson p_hampson@wargaming.net](#) Wargaming N4109 A proposal to add a utility class to represent expected monad
- [2.3.6 Massive Parallelism with Heterogeneous devices \(Michael Wong, Lee Howes\)](#)
 - [Michael Wong fraggamuffin@gmail.com](#) ISOCPP Towards Massive Parallelism(aka Heterogeneous Devices/Accelerators/GPGPU) support in C++ , Khronos's OpenCL SYCL to support Heterogeneous Devices for C++
- [2.3.7 help better support latency in financial/trading/banking applications \(TBD\)](#)

Around the same time, EASTL was announced by Roberto Paolin to be now released on github. This was the original paper that was written by Paul Padriana back around 2007/8 when we were almost too busy to pay attention to anything else while trying to release C++11. We have come a long way and the C++ committee is definitely listening.

Status

Current papers working through WG21, proposed for the Oulu meeting.

1. flat map: Sean M
2. fixed point: John M, Lawrence, Marco Foco
3. ring span: Guy, Arthur
4. Low level bit manip: Vincent
5. uninit memory algo: Brent Friedman
6. datapar (SIMD) : M Kretz, M Gaunard, Joel Falcou
7. Comparing virtual fns: Scott Wardle/Sunil
8. Thread constructor attributes: Patrice
9. install distribution packaging: Brian Fitzgerald
10. unstable move: Brent
11. Class for status/Optional: Lawrence
12. utility class to represent expected monad: Paul Hampson, Vincent Botet
13. hazard pointers and RCU for lock-free programming: Michael
14. 2d display: Michael M
- 15 massive dispatch for heterogeneous devices: Michael, Hartmut, Lee
16. Lock-free concurrency tool-kit using hazard pointers and RCU: Michael, Maged, Paul
17. Memory ordering: consume ordering: Paul, Michael

Status of future SG14 proposals and their owner. This is a listing of proposals actively discussed as recently as 2016 on the reflector. There may be earlier proposals which are not reflected here.

1. Finance/Trading thread and recent STAC SG14 meetings
<https://groups.google.com/a/isocpp.org/forum/#!topic/sg14/4WvbE2iaFNI>
<https://groups.google.com/a/isocpp.org/forum/#!topic/sg14/0em3tc5uuwI>

These meetings have identified a number of main features

- a. Heterogeneous device support (Michael, Hartmut)
- b. EH lite (see below)
- c. CPU/cache/memory affinity (Neil)
- d. memory allocation (Guy Davidson)
2. multiple small vectors: Gonzalo BG et al
3. Exception lite and swift-like exceptions: Sunil, Patrice, and Sean Middleditch
4. basic inplace function: Nicolas Fleury, Sean M, etc al
5. Interprocess communication: Shaun Croton et al
6. hot set: Brent Friedman
7. Accessors: Andrew, Ronan, Lee
8. std::stack: Matthew Bentley
9. plf::colony/stack: Matthew Bentley
10. FAst associative container: Allan Deutsch
11. Alternatives to traverse linked data structures: Marcelo Zimbres
12. Delegates for simulations: Miodrag Milanovic
13. Dynamic/runtime concepts: Zach Laine, Andrew Sutton
14. width/set width: John M
15. explicit initializer list constructors: Nicolas Fleury
16. popping move-only types from priority queue: Ben Deane
17. affinity, locality and hints: Neil Horlock
18. intrusive containers (Guy)
19. FPGAs (Ronan)
20. lock-free queues: michael/lawrence
21. half precision types: Sean M
22. likely/unlikely: Bartosz

If you are interested in following these proposals, please join the discussion.

Future Meetings

1. June 13 STAC NY, looking for a host for Tues June 14th: 10-11:30
2. Monday June 27: HFT for C++ Amsterdam hosted by Optiver, Carl Cook
3. CPPCON 2016: Wed, Sept 21: 8:30-5 pm
4. Meeting C++ Games Track: Nov 18-19, 2016, Berlin

Future Directions

There will be even more specific focus on Low latency given the even more important real-time requirements of VR, simulation, HFT, massive parallelism in heterogeneous dispatch, AI and deep computer learning with neural network, and self-driving cars. The major long-pole items that could be gathered into an SG14 TS include Affinity, Process vs threads, EH lite and new allocators. The demand for heterogeneous and low latency devices is becoming pervasive and many such as finance, trading, games, and data centre already are major C++ users.

A major future drive will be to continue connection with heterogeneous device programming as well as financial, and further connect more with Embedded computing members as well as that of big data.

References

N4456: Towards improved support for games, graphics, real-time, low latency, embedded systems;

<http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2015/n4456.pdf>

P0364R0 Report on Exception Handling Lite (Disappointment) from SG14